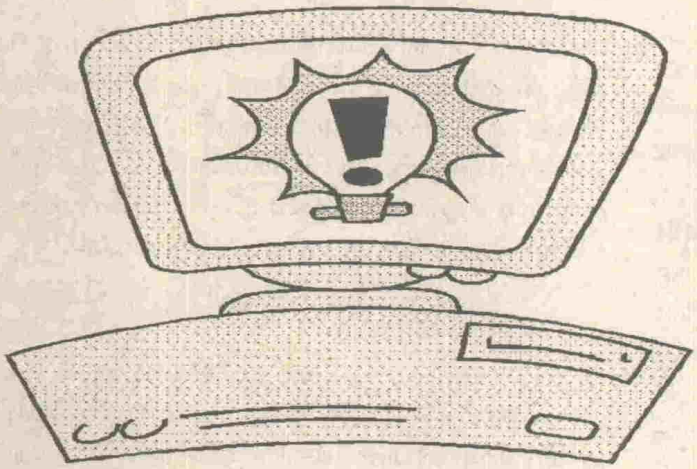
Classical Windows(TM) Programming : การซับคลาสซิ่งในวินโดวส์กับการโอเวอร์ไรด์ในภาษาเชิงวัตถุ



ถ้าคุณต้องการสร้างกรอบอีดิตที่รับค่าเฉพาะตัวเลข ไม่รับตัวอักษรหรือสัญลักษณ์อื่น จะสามารถสร้างอีดิตนี้ได้อย่างไร

วันเวลาผ่านไปเร็วเหมือนความฝัน อะไรหลาย ๆ อย่างเปลี่ยนไปในทางที่ดีขึ้น เทคโนโลยีของคอมพิวเตอร์ทั่วโลกยิ่งพัฒนาไปเร็วยิ่งกว่า ไม่ว่าจะเป็นฮาร์ดแวร์ ซอฟต์แวร์ หรือพีเพิลแวร์ (peopleware) ยกเว้นเพียงในประเทศเดียวเท่านั้นในโลกที่เทคโนโลยีทางคอมพิวเตอร์ของประเทศนั้นพัฒนาไปช้าพอ ๆ กับการพัฒนาทางการเมืองของพวกเขา แม้ว่าจะมีรัฐธรรมนูญที่วิเศษขนาดไหนก็ตาม

ในการเขียนโปรแกรมก็เปลี่ยนไป จากแต่ก่อนเขียนโปรแกรมเป็นภาษาเครื่อง เมื่อมีเครื่องรุ่นใหม่เข้ามาก็ต้องเขียนโปรแกรมกันใหม่ ต่อมาจึงมีผู้คิดระบบปฏิบัติการและคอมไพเลอร์ขึ้นมา เครื่องที่ใช้สถาปัตยกรรมเดียวกันก็สามารถใช้ระบบปฏิบัติการเดียวกันได้ ไอบีเอ็มประสพความสำเร็จอย่างสูงกับแนวคิดนี้

จากนั้นมีผู้คิดไมโครโปรเซสเซอร์ขึ้นมา อินเทลจึงถือกำเนิดขึ้น การใช้ไมโครโปรเซสเซอร์เป็นที่นิยมกันอย่างแพร่หลาย มีวัยรุ่นสองคนช่วยกันเขียนซอฟต์แวร์อินเทอร์พรีเตอร์ (Interpreter) ภาษาเบสิกขึ้นมา และเป็นจุดเริ่มต้นของไมโครซอฟท์ เด็กสองคนที่ว่านั้นคือ พอล แอเล็น กับ บิลล์ เกตส์

เมื่อคอมพิวเตอร์ส่วนบุคคลแพร่หลาย แพล็ตฟอร์ม (platform) ของเราจึงเปลี่ยนไป เราเขียนโปรแกรมกันบนดอส (DOS(TM)) บนยูนิกซ์ (UNIX (TM)) ยุคต่อมาเมื่อคอมพิวเตอร์ที่มีจอกราฟิกมีราคาถูกลงแพล็ตฟอร์มเราเปลี่ยนเป็นวินโดวส์ (Windows(TM)) ซึ่งในขณะนั้นเป็นเชลล์ของดอส หรือ X-Windows ก็เป็นเชลล์ของยูนิกซ์หรือจะเป็นแมคอินทอช (Macintosh(TM)) ซึ่งในตอนนั้นเราเขียนโปรแกรมเพื่อจัดการกับเหตุ (Event Driving)

ต่อมาเมื่อการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming - OOP) เริ่มได้รับความนิยม แนวคิดของการนำซอฟต์แวร์กลับมาใช้ใหม่ (Reusability Software) เข้ามามีบทบาท ประกอบกับวิชวลโปรแกรมมิ่ง (Visual Programming) ก็มีลักษณะคล้ายกับประดิษฐ์สิ่งของต่าง ๆ ตามในชีวิตจริง ไมโครซอฟท์ได้แนะนำวิชวลเบสิกและคอนโทรลเพิ่มเติม (VBX) เพื่อสร้างตลาดให้ผู้ผลิตคอนโทรล ต่อมามีผู้ผลิตวิชวลโปรแกรมมิ่งในหลายภาษา ไม่ว่าจะเป็น Borland Delphi, Borland C++ Builder, Powersoft PowerBuilder ฯลฯ

เมื่อเน็ตเวิร์กเข้ามามีบทบาทก็มีคอนโทรลหรือคอมโปเนนต์ในทางเน็ตเวิร์กมากขึ้น ซันไมโครซิสเต็มส์ได้ส่งภาษาจาวา Java(TM) มาแข่งขัน โดยมีการแจกคอมไพเลอร์ฟรี โดยจาวามีลักษณะพิเศษคือ สามารถ

Classical Windows(TM) Programming : การซับคลาสซิ่งในวินโดวส์กับการโอเวอร์ไรด์ในภาษาเชิงวัตถุ

ทำงานได้ทุกแพล็ตฟอร์ม แต่ต้องรันบนแพล็ตฟอร์มมาตรฐานของตัวเองที่เรียกว่า Java Virtual Machine การทำงานกับเน็ตเวิร์กก็ทำได้ง่าย ยูสเซอร์อินเทอร์เฟสก็สวยงาม มีการจัดการฐานข้อมูล และซันยังได้สร้างตลาดของคอมโปเนนต์เช่นเดียวกันคือ จาวาบีน (JavaBean(TM))

หากจะเปรียบการเขียนโปรแกรมกับงานศิลปะเหมือนเช่นท่านปรมาจารย์แห่งการเขียนโปรแกรม ท่าน Donald E. Knuth มักจะชอบเปรียบเปรย ในศิลปะมีหลายยุค ที่พอจะดูรู้เรื่องก็เริ่มจาก Egyptian, Roman เข้าสู่ยุค Renaissance ซึ่งในยุคนี้เองก็เกิดชื่อดังมากมาย เช่น เลโอนาร์โด ดาร์วินชี, ไมเคิล แองเจโล, ราฟาเอล (ศิลปินครับ ไม่ใช่นินจาเต่า) จากนั้นเกิดการปฏิวัติที่ฝรั่งเศส เกิดศิลปะยุคใหม่คือนีโอ คลาสสิก (Neo Classic) ก็เกิดสไตล์ของงานศิลปะมากมายจนถึงปัจจุบัน ไม่ว่าจะเป็น realism, impressionism, surrealism, abstract ฯลฯ

ศิลปินที่มีความชอบในสีที่ต่างกัน (เช่น บางคนชอบสีน้ำ บางคนชอบสีน้ำมัน) เปรียบเหมือนกับโปรแกรมเมอร์ที่ชอบในภาษาที่ต่างกัน บางคนชอบเขียนโปรแกรมบนยูนิกซ์ บางคนเขียนบนวินโดวส์ บางคนเขียนเป็น CGI (Common Gateway Interface) ซึ่งรันบนเน็ตเวิร์ก

กลับมาที่เรื่องการเขียนโปรแกรมครับ ไมโครซอฟท์ได้แนะนำวินโดวส์ในปี 1983 ซึ่งในขณะนั้นแนวคิดเรื่องการเขียนโปรแกรมเชิงวัตถุยังไม่เป็นที่ยอมรับ ไมโครซอฟท์จึงได้เลือกใช้ภาษาอ้างอิงในการเขียนโปรแกรมบนวินโดวส์คือ ภาษาซี ซึ่งในบทความนี้ผมจึงขอเสนอเทคนิคในการเขียนโปรแกรมบนวินโดวส์ ซึ่งหลายเทคนิคแทบจะไม่มีคนรู้จัก (อาจเป็นเพราะมีวิธีอื่นที่สะดวกกว่า) แต่เพื่อความสะดวก (ของผม) และเพื่อให้มีข้อแตกต่างกับนักเขียนท่านอื่นผมจะใช้ภาษาซีในการนำเสนอ

Subclassing

หากคุณสังเกตให้ดี ในอีดิตคอนโทรลของวินโดวส์ฉบับภาษาไทยนั้นตัวอักษร 2 ตัวที่อยู่ติดกัน หากไม่สามารถประกอบเป็นคำได้ คุณจะไม่สามารถพิมพ์ได้ เช่น คำว่า "น้ำ" จะต้องพิมพ์ไม้โทก่อนสระอำ (ยกเว้นในไมโครซอฟท์เวิร์ด) เพราะในภาษาไทยนั้นหากคุณพิมพ์ข้อมูลสะเปะสะปะแล้วโปรแกรมบางตัวจะไม่สามารถทำงานบางอย่าง เช่น การเรียงข้อมูลได้ถูกต้อง แต่ในวินโดวส์ภาษาอังกฤษนั้นคุณจะพิมพ์อย่างไรก็ได้ หากคุณเป็นโปรแกรมเมอร์ที่ได้รับมอบหมายงานให้ทำสิ่งนี้ เพราะคุณเป็นเพียงคนเดียวที่รู้เรื่องภาษาไทยดีกว่าโปรแกรมเมอร์ฝรั่ง คุณจะทำอย่างไร อย่าลืมว่าคุณไม่สามารถเข้าถึงการทำงานในระดับแก่น (Kernel) ได้

อีกปัญหาหนึ่ง หากคุณต้องการสร้างกรอบอีดิตที่รับค่าเฉพาะตัวเลข ไม่รับตัวอักษรหรือสัญลักษณ์อื่น จะสามารถสร้างอีดิตนี้ได้อย่างไร

โปรแกรมเมอร์รุ่นใหม่คงจะตอบตามสไตล์ของโอโอพี (และวิชวลโปรแกรมมิ่ง) ว่าสิ่งเหล่านี้ทำได้ง่ายมาก นั่นคือการโอเวอร์ไรด์ (Override เข้าควบคุม) การทำงานของอีดิตคอนโทรลนั้นเพื่อให้มันทำงานอย่างที่เราต้องการ แต่ช้าก่อน! คุณอย่าลืมว่า ตัววินโดวส์เองนั้นเขียนขึ้นมาด้วยภาษาซี ซึ่งไม่ใช่ภาษาเชิงวัตถุ ฉะนั้นทางออกของภาษาซีคือซับคลาสซิ่ง (Subclassing)

ซับคลาสซิ่งนี้คิดค้นมาจากศูนย์วิจัยซีรอกซ์ปาร์ค ซึ่งเป็นผู้คิดค้นและวิจัยระบบการทำงานแบบวินโดว์เป็นการสั่งให้คอนโทรล (หน้าต่าง) ที่กำหนดมีพฤติกรรมต่างจากเดิมหรือเพิ่มจากเดิมก็ได้ ซึ่งคล้ายกับการโอเวอร์ไรด์ในภาษาเชิงวัตถุ (Inherited ใน Pascal และ super ใน Java)

ทุกวินโดว์ (คอนโทรล) ที่มีแฮนเดิล จะมีวินโดว์โพรซีเดอร์ของตัวเองเสมอ เมื่อเราได้ค่าแฮนเดิลของวินโดว์มา (เช่น จาก GetDlgItem) เราจะหาแอดเดรสของวินโดว์โพรซีเดอร์ที่ต้องการได้ด้วยฟังก์ชัน GetWindowLong ซึ่งมีรูปแบบการใช้คือ

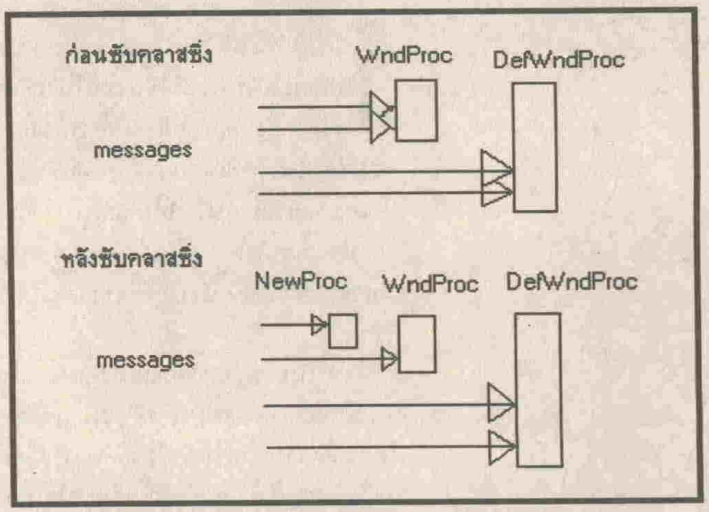
```
LONG GetWindowLong(HWND hWnd, int nIndex)
```

โดยที่ hWnd คือแฮนเดิลของวินโดว์ที่เราต้องการ nIndex คือค่าที่เราต้องการทราบ ในที่นี้คือแอดเดรสของวินโดว์โพรซีเดอร์ เราจะใช้ GWL_WNDPROC ส่วนค่าอื่นขอให้ศึกษาในคู่มือ

เราจะเก็บค่าแอดเดรสนี้ไว้ก่อนเพื่อนำมาใช้ภายหลังคล้าย ๆ การ paint ในวินโดว์ จากนั้นเราจะทำการเปลี่ยนวินโดว์โพรซีเดอร์ของวินโดว์ที่ต้องการด้วยฟังก์ชัน SetWindowLong มีรูปแบบคือ

```
BOOL SetWindowLong(HWND hWnd, int nIndex,
LONG dwNewLong)
```

ค่า dwNewLong คือค่าใหม่ที่ต้องการเซต ซึ่งในกรณีนี้คือแอดเดรสของวินโดว์โพรซีเดอร์ใหม่ หากเราเขียนวินโดว์โพรซีเดอร์ธรรมดา เมสเสจที่เราไม่สนใจเราจะส่งให้ดีฟอลต์ (เช่น DefWindowProc, DefDlgProc, DefScreenSaverProc) แต่การซับคลาสซิ่งนี้เราควรจะให้วินโดว์โพรซีเดอร์เดิมจัดการและหาวินโดว์โพรซีเดอร์เดิมไม่ต้องการก็จะส่งให้ดีฟอลต์ต่อไป (เหมือนไม่มีอะไรเกิดขึ้น) ในรูปที่ 1

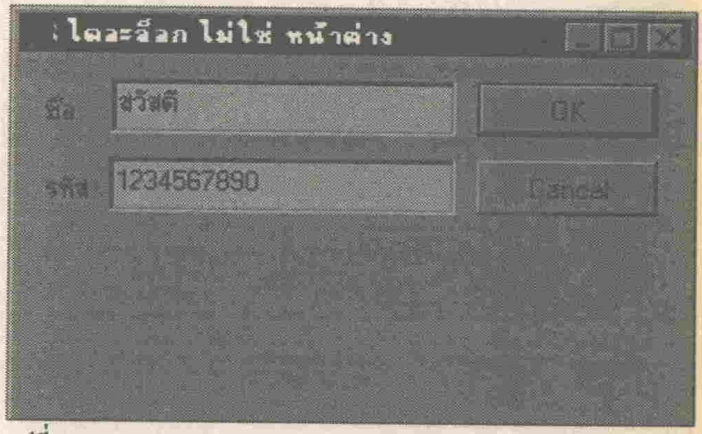


รูปที่ 1 แสดงการไหลของเมสเสจทั้งก่อนการซับคลาสซิ่งและหลังจากซับคลาสซิ่งแล้ว

Classical Windows(TM) Programming
: การซับคลาสซิ่งในวินโดวส์กับการโอเวอร์ไรด์
ในภาษาเชิงวัตถุ

โปรแกรม Magic

โปรแกรมตัวอย่างแสดงในลิสติงที่ 1 เพื่อให้ดูสั้นและตรงกับหลักการที่อธิบายข้างต้น ผมจึงใช้ไดอะล็อกเป็นวินโดว์หลัก และได้สร้างไดอะล็อกจากรีซอร์ส (ใช้เอดิเตอร์ หรือพิมพ์เข้าไปเลยก็ได้) ตัวอีดิตที่ถูกซับคลาสคือ IDC_EDIT2 ซึ่งจะรับข้อมูลที่เป็นตัวเลขเท่านั้น ผลการรันที่ได้เป็นดังในรูปที่ 2 สังเกตในฟังก์ชัน NumberProc() เมสเสจที่ไม่สนใจเราส่งให้วินโดว์โพรซีเดอร์เดิมด้วย CallWindowProc() (พารามิเตอร์ตัวแรกเป็นแอดเดรสของวินโดว์โพรซีเดอร์ และสี่ตัวหลังเหมือนกับวินโดว์โพรซีเดอร์ทั่วไป)



รูปที่ 2

ลิสติงที่ 1

```
/*
 * Magic.c
 *
 * by Kookiat Suetrong (c)
 */

#include
#include
#include "Resource.h"

FARPROC lpfnOldProc;

BOOL WINAPI NumberProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
        char ch;

        switch (uMsg) {
        case WM_CHAR :
                ch = (TCHAR)wParam;
                if (ch >= '0' && ch <= '9') {
                        return CallWindowProc(lpfnOldProc, hWnd, uMsg, wParam, lParam);
                }
                return TRUE;
                break;
        default :
                return CallWindowProc(lpfnOldProc, hWnd, uMsg, wParam, lParam);
                break;
        }
}

BOOL WINAPI DlgProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
        BOOL bRet = TRUE;
        HWND hEdit;

        switch (uMsg) {
        case WM_INITDIALOG :
                hEdit = GetDlgItem(hWnd, IDC_EDIT2);
                lpfnOldProc = (FARPROC)GetWindowLong(hEdit, GWL_WNDPROC);
                SetWindowLong(hEdit, GWL_WNDPROC, (LONG)MakeProcInstance((FARPROC)NumberProc, hWnd));
                bRet = TRUE;
                break;
        case WM_COMMAND :
                if (GET_WM_COMMAND_ID(wParam, lParam) == IDCANCEL)
                        EndDialog(hWnd, 0);
                break;
        default :
```

→

ลิสติงที่ 1 (ต่อ)

```
                bRet = FALSE;
                break;
        }
        return bRet;
}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
{
        DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG1), NULL, DlgProc);

        return 0;
}


// resource.h
#define IDD_DIALOG1                     101
#define IDC_EDIT1                       1001
#define IDC_EDIT2                       1002


// Magic.rc
#include "resource.h"
#include "afxres.h"

LANGUAGE LANG_NEUTRAL, SUBLANG_NEUTRAL

IDD_DIALOG1 DIALOG DISCARDABLE  0, 0, 185, 95
STYLE WS_MINIMIZEBOX | WS_MAXIMIZEBOX | WS_CAPTION | WS_SYSMENU |
    WS_THICKFRAME
CAPTION "ไดอะล็อก ไม่ใช่ หน้าต่าง"
FONT 8, "MS Sans Serif"
BEGIN
    PUSHBUTTON      "OK",IDOK,127,6,50,14
    PUSHBUTTON      "Cancel",IDCANCEL,127,27,50,14
    EDITTEXT        IDC_EDIT1,25,6,95,14,ES_AUTOHSCROLL
    EDITTEXT        IDC_EDIT2,25,27,95,14,ES_AUTOHSCROLL
    LTEXT           "ชื่อ",IDC_STATIC,7,10,10,8
    LTEXT           "รหัส",IDC_STATIC,7,31,14,8
END
```

ก่อนจบ

การซับคลาสซิ่งยังขาดอะไรบางอย่าง คือในขณะที่เราซับคลาสคอนโทรล เราจะทำได้เพียงแค่ครั้งละตัวเดียวเท่านั้น หากจะทำให้เหมือนภาษาเชิงวัตถุจริง ๆ แล้วจะต้องใช้เทคนิคที่เรียกว่า ซุปเปอร์คลาสซิ่ง ซึ่งหากมีโอกาสผมจะนำมาเล่าให้ฟัง

บทความนี้น่าจะเป็นประโยชน์มากสำหรับการใช้งานในบางกรณี (เช่น เกี่ยวกับภาษาไทย ทั้งรับและแสดงผล) ซอร์สโค้ดที่นำเสนอใช้ภาษาซี สามารถคอมไพล์ได้กับคอมไพเลอร์ทุกตัว ซึ่งเป็นเรื่องที่น่าสนใจ แต่หลายคนมองข้ามคอมไพเลอร์ภาษาซี (และซีพลัสพลัส) นั้นมีประสิทธิภาพมากกว่าภาษาอื่นหลายเท่าตัว กลับมีราคาถูกกว่าคอมไพเลอร์ภาษาอื่นอย่างไม่น่าเชื่อ ผู้เขียนไม่ทราบว่าผู้อ่านมีพื้นฐานของวินโดวส์ API มากเพียงใด

หากต้องการให้เขียนเกี่ยวกับเรื่องนี้อีก ขอให้ผู้อ่านช่วยเขียนมาแนะนำได้ที่ beerbeer@geocities.com หากมีคำถามเกี่ยวกับเรื่องของ Microsoft Foundation Class (MFC) ถามได้ที่เดียวกัน แล้วพบกันใหม่สวัสดี